Predicting Redshifts Using Random Forests

Amir Kazemi-Moridani

Outline

- Random forests
- Our data
- Pre-processing
- Optimization and results

Schedule
========
Week 1:
- Get the data and make some preliminary plots (axis ratio, size, photometry)
- Investigate what packages are available for random forest
- Stretch goal: Make a tree (or a series of trees)

Week 2:
- Generate the random forest fits for a series of galaxies (probably just using photometry)
- Evaluate goodness of fit

Week 3:
- Parameterize the uncertainties of the fits using the trees

Week 4:
- Explore other parameters (such as size, axis ratio)

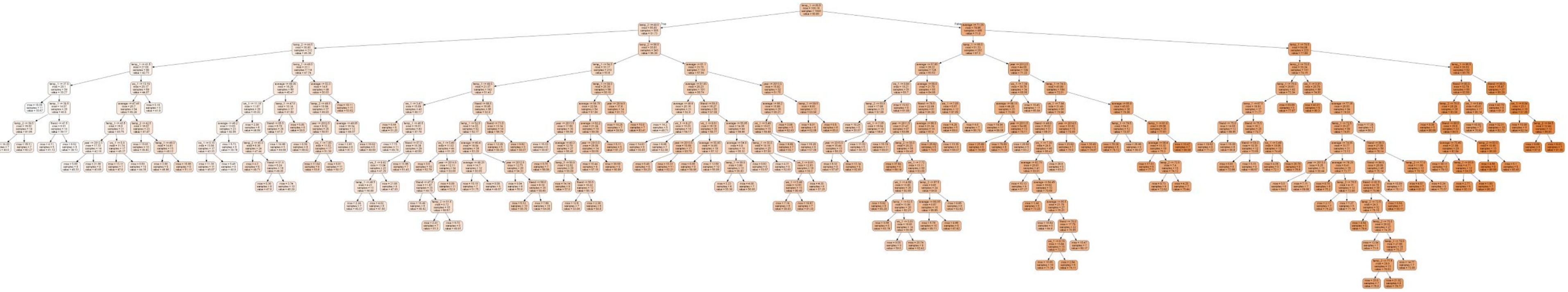
Week 5:
- Finalize the fitting parameters
- Potentially vary the number of photometric data points in the fit to see the extent to which extra parameters aid the fit

Week 6: DESC Meeting
- Finalize fits

Week 7: Thanksgiving
- Summarize results of adding different numbers of parameters

Random Forests

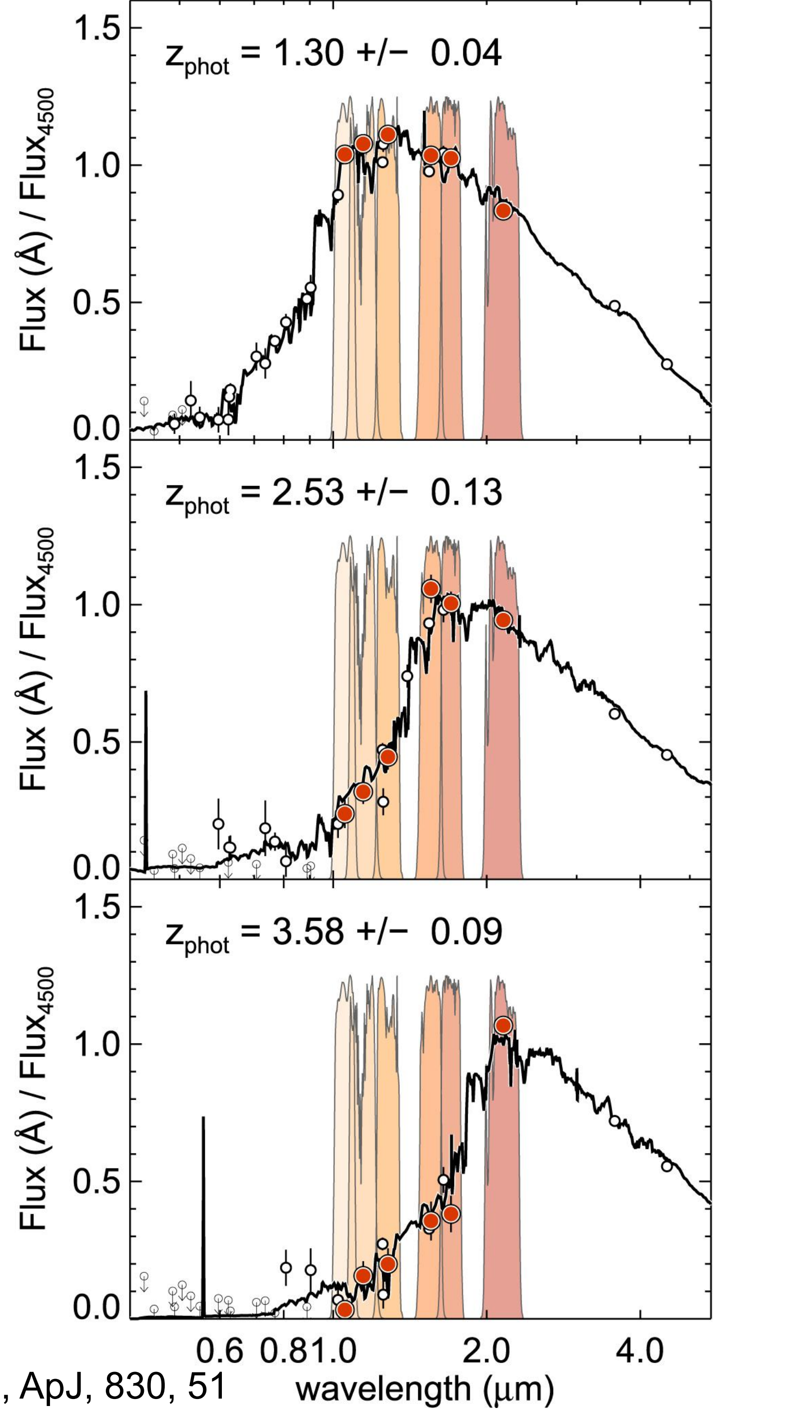
- Forests of prediction trees
  - Decision trees - discrete
  - Regression trees - continuous
- Each tree is constructed by selecting the best split point from a **random** subsample of the dimensions - minimizing the sum of squared errors
  - Trained on bootstrap samples of the data
  - Using a subset of features
  - Terminated based on provided criteria (e.g. minimum leaf size)



https://towardsdatascience.com/hyperparameter-tuning-the-random-forest-in-python-using-scikit-learn-28d2aa77dd74

Photometric Redshift (photo-z)

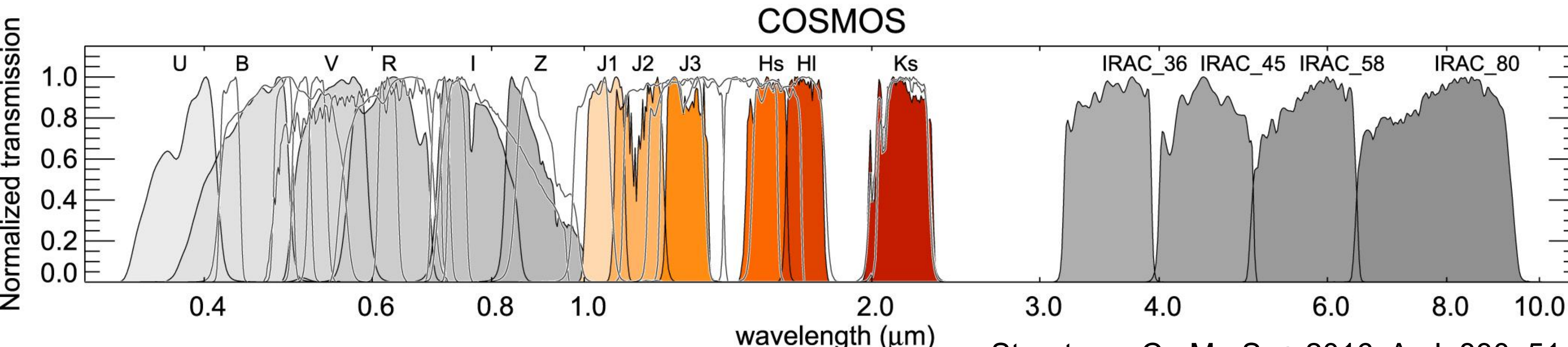
- Spectroscopic surveys are expensive and time-consuming
- Photometric measurements are faster and less expensive



Straatman C.~M.~S. + 2016, ApJ, 830, 51

The Data

- From the FourStar galaxy evolution survey (ZFOURGE)



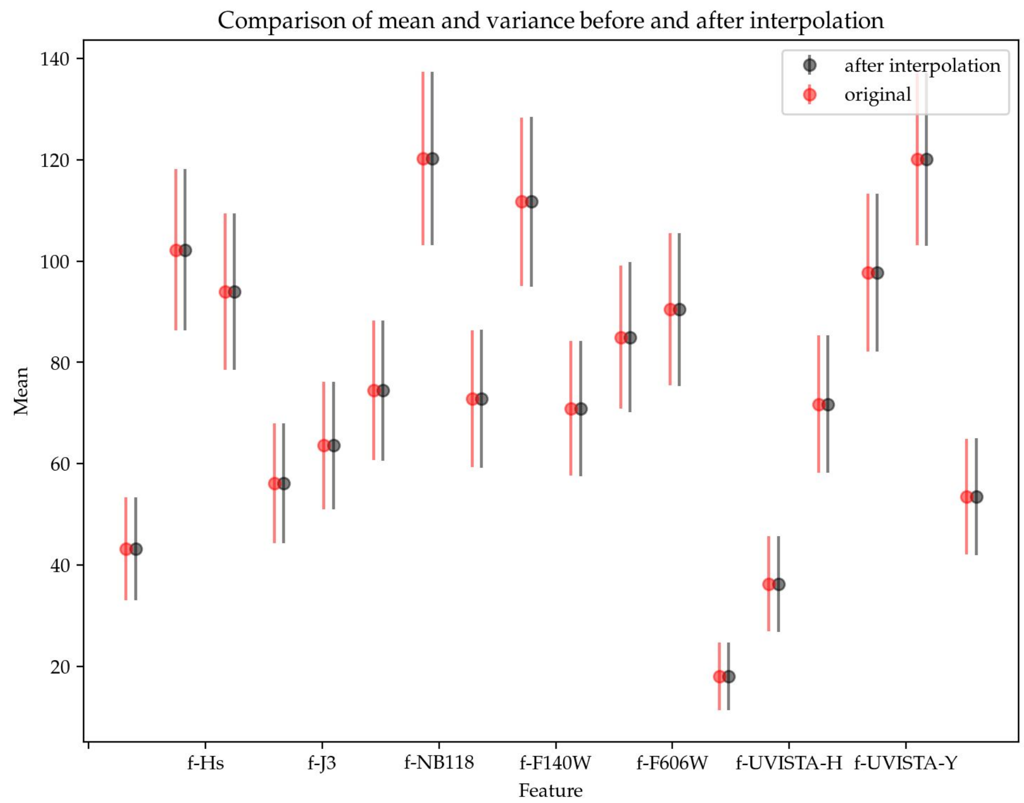
Straatman C.~M.~S. + 2016, ApJ, 830, 51

	id	x	y	ra	dec	SEflags	iso_area	fap_Ksall	eap_Ksall	apcorr	...	wmin_jhk	wmin_hst	wmin_irac	wmin_all	star	nearstar
0	166	3781.674	109.945	150.088715	2.177286	0	62.0	1.396826	0.116562	1.143497	...	1.08	0.00	1.01	0.54	0	0
1	553	4079.608	229.830	150.076294	2.182282	2	252.0	7.455316	0.116562	1.099861	...	1.02	0.00	1.05	0.54	0	0
2	641	3734.180	297.363	150.090698	2.185095	3	63.0	1.096613	0.116562	1.119066	...	1.00	0.00	0.94	0.54	0	0
3	658	3406.160	300.911	150.104370	2.185241	3	75.0	1.203554	0.121134	1.078640	...	1.01	0.81	0.90	0.50	0	0
4	668	3777.161	294.776	150.088898	2.184987	2	114.0	1.680287	0.116562	1.126751	...	1.01	0.00	0.94	0.54	0	0

	f_Ksall	f_B	f_G	f_I	f_IA427	f_IA484	f_IA505	f_IA527	f_IA624	f_IA709	...	f_F606W	f_F814W	f_UVISTA_J	f_UVISTA_H
0	2.763520	0.803838	0.938691	1.133950	0.540159	0.815649	1.124296	0.950815	1.166044	1.100923	...	1.078534	1.042126	1.644773	2.375077
1	19.479900	3.107052	3.531762	9.960004	2.531886	3.616404	3.814700	4.016532	8.534525	9.938734	...	6.434791	10.598488	15.004392	17.988017
2	2.881431	0.680146	0.775524	0.980625	0.557790	0.817122	0.639527	0.788392	1.045351	1.008576	...	0.837689	1.104082	1.947452	3.042136
3	3.957254	1.669803	1.714664	1.723101	1.660514	1.693340	1.721699	1.603944	1.748214	1.548854	...	1.717241	1.768363	2.710785	4.566974
4	3.965060	1.322258	1.427895	1.668771	1.237378	1.355434	1.501069	1.479561	1.600855	1.710845	...	1.393661	1.460127	3.348458	4.078588

Pre-processing

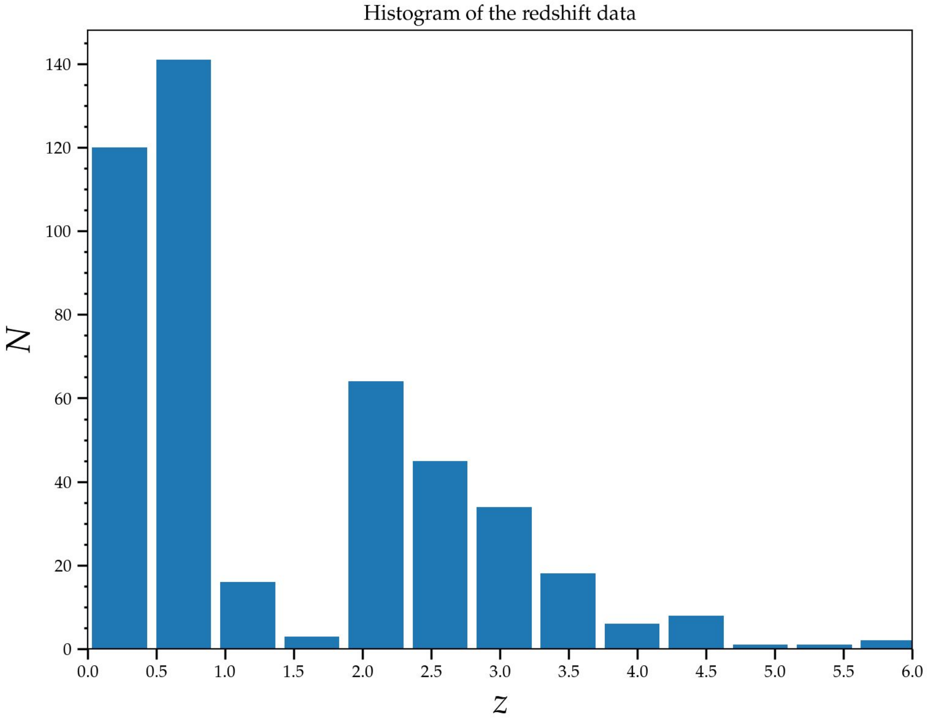
- Eliminating columns that are not flux measurements
- Eliminating objects that might be stars
- Interpolating to estimate the missing values
  - Using a kNN algorithm (using ***impyute*** package)

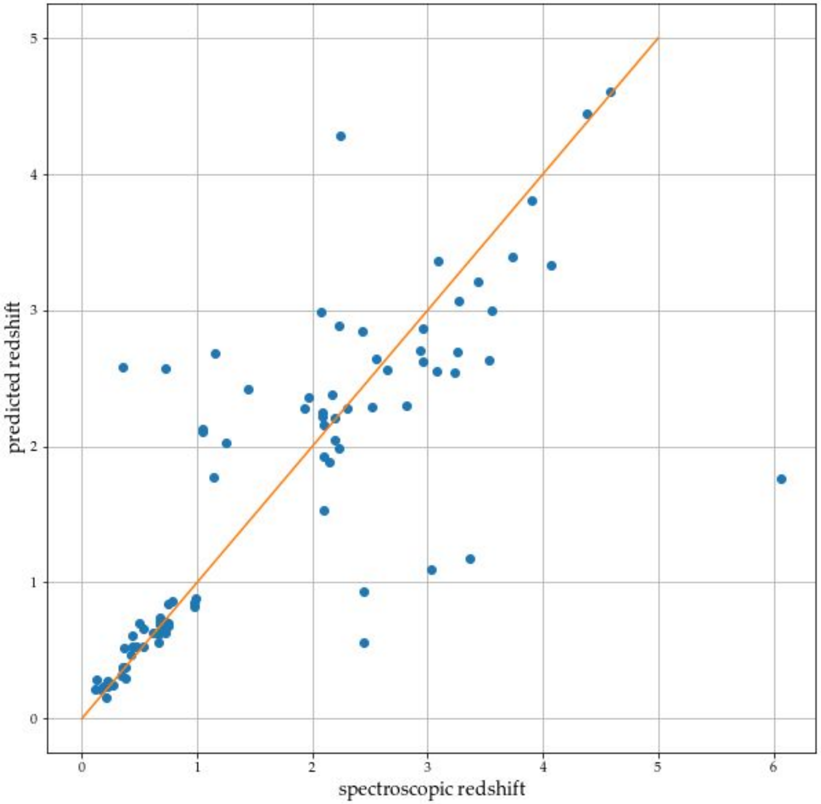



```
f_H1 is missing 1 values
f_Hs is missing 1 values
f_J1 is missing 1 values
f_J2 is missing 2 values
f_J3 is missing 1 values
f_Ks is missing 1 values
f_NB118 is missing 19 values
f_NB209 is missing 21 values
f_F125W is missing 8 values
f_F140W is missing 75 values
f_F160W is missing 3 values
f_F606W is missing 2 values
f_F814W is missing 2 values
f_UVISTA_Ks is missing 5 values
f_UVISTA_Y is missing 5 values
```

Initial Results

- Results with default arguments of Scikit learn random forest regressor





Optimizing the Hyperparameters

- Parameters of the regressors

```
{'bootstrap': [True, False],
 'criterion': ['mae', 'mse'],
 'max_depth': [20, 36, 52, 69, 85, 101, 118, 134, 150, 167, 183, 200, None],
 'max_features': ['auto', 'sqrt'],
 'min_samples_leaf': [1, 2],
 'min_samples_split': [2, 5, 10, 15, 20],
 'n_estimators': [100, 129, 166, 215, 278, 359, 464, 599, 774, 1000]}
```

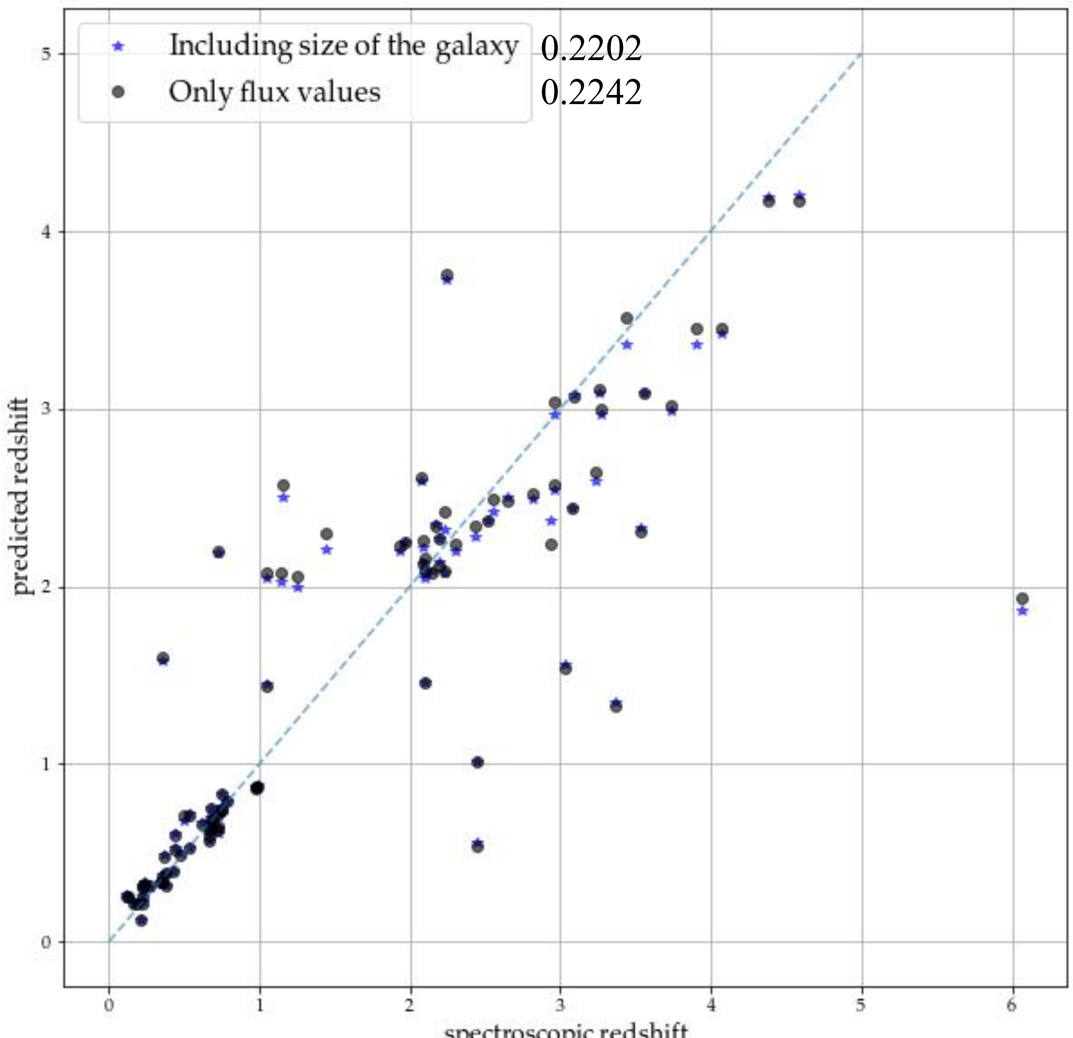
- Scikit learn packages
    - RandomizedSearchCV - ~4 hrs of runtime
    - GridSearchCV - ~3 hrs of runtime

Final Results

f_I	:	0.1089
f_Zp	:	0.0974
f_Z	:	0.0772
f_IA738	:	0.0648
f_UVISTA_J	:	0.0626
f_UVISTA_Y	:	0.0485
f_J1	:	0.0468
f_IRAC_45	:	0.0337
f_IA709	:	0.0316
f_J3	:	0.0313
f_U	:	0.0289
f_Rp	:	0.0274
f_J2	:	0.0258
f_F125W	:	0.0233
f_IRAC_36	:	0.0206
f_IA505	:	0.0196
f_V	:	0.0192
f_IA624	:	0.0187
f_G	:	0.0184
f_IA527	:	0.0182
f_F814W	:	0.018
f_IA427	:	0.0166
f_IA484	:	0.0163
f_B	:	0.016
f_R	:	0.013
f_F606W	:	0.0122
f_IRAC_58	:	0.0115
f_Hs	:	0.0107
f_F160W	:	0.01
f_NB118	:	0.0098
f_H1	:	0.0095
f_Ks	:	0.0058
f_UVISTA_H	:	0.0056
f_IRAC_80	:	0.0054
f_Ksall	:	0.0048
f_UVISTA_Ks	:	0.0042
f_NB209	:	0.0039
f_F140W	:	0.0039

f_Zp	:	0.0898
f_I	:	0.0864
f_Z	:	0.0796
f_IA738	:	0.0645
f_UVISTA_J	:	0.0566
f_J1	:	0.0522
f_UVISTA_Y	:	0.0385
f_IA709	:	0.0345
f_IRAC_45	:	0.0332
f_F814W	:	0.0323
f_J2	:	0.0295
f_J3	:	0.0271
f_U	:	0.0262
f_Rp	:	0.024
f_IA624	:	0.0232
f_F125W	:	0.0219
f_IRAC_36	:	0.0193
f_G	:	0.019
f_IA427	:	0.0183
f_IA505	:	0.018
f_V	:	0.0172
f_R	:	0.0169
f_IA527	:	0.0163
f_IA484	:	0.016
f_B	:	0.0159
f_F160W	:	0.0159
f_F606W	:	0.0122
f_IRAC_58	:	0.0112
f_Hs	:	0.0108
f_NB118	:	0.0103
f_H1	:	0.0094
f_Ksall	:	0.0085
b_vector	:	0.0078
a_vector	:	0.0064
f_UVISTA_H	:	0.0064
f_UVISTA_Ks	:	0.0059
f_Ks	:	0.0057
f_IRAC_80	:	0.0049
f_F140W	:	0.0039
f_NB209	:	0.0038

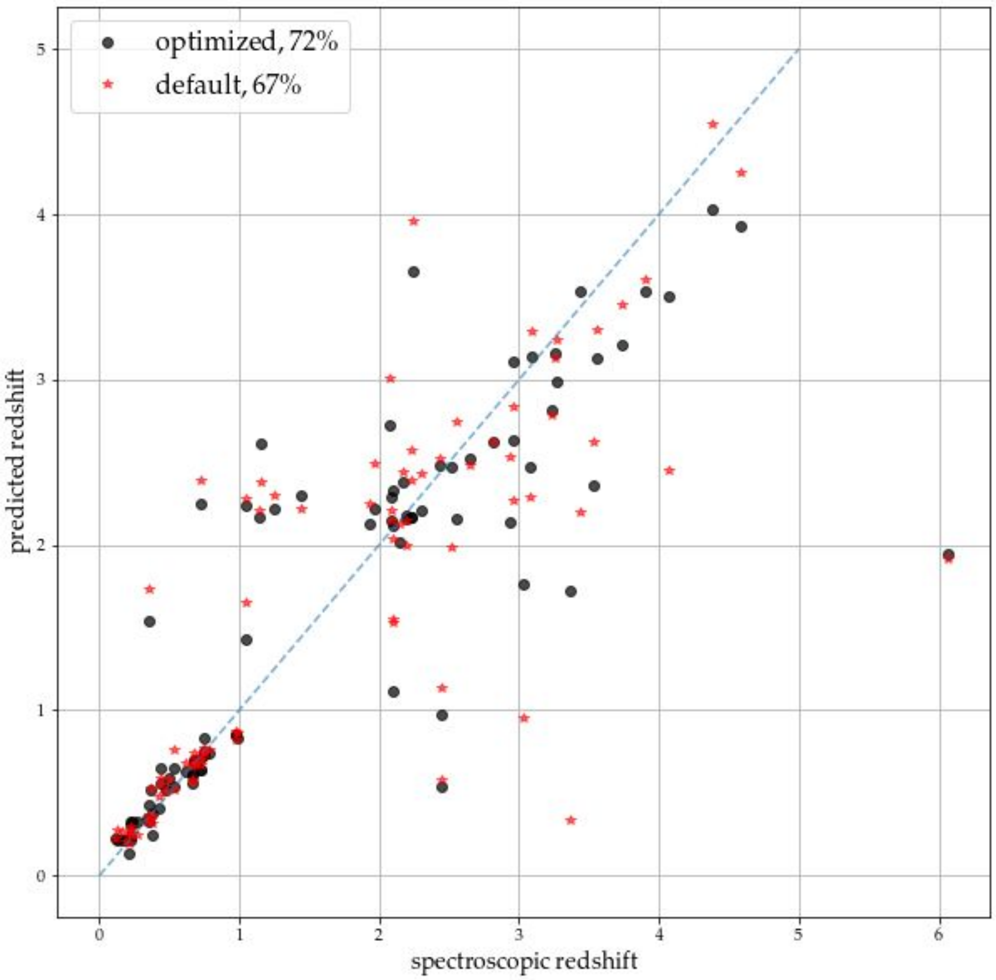
$$\mathrm{rms}\left(\frac{\Delta z}{1+z}\right)$$



Final Results

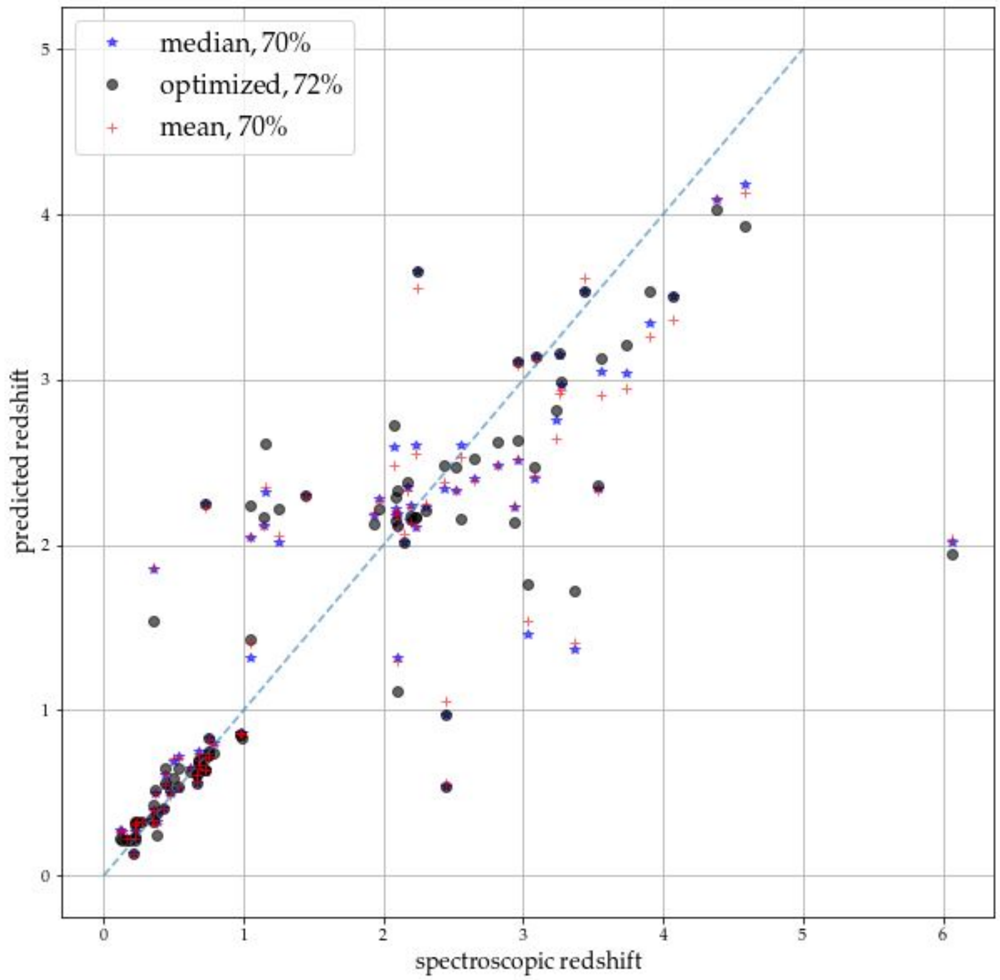
- Accuracy =

$$1 - \frac{1}{N}\sum_{i}\frac{|z_{\text{rf, i}} - z_{\text{spec, i}}|}{z_{\text{spec, i}}}$$



Final Results

- Making 5 sets for cross validation and fitting 5 random forests
- Predicting the redshift using each tree
- Taking the median or mean of the 5 values as the estimate



Summary

- Random forests are robust and easy to implement
- Imputation is crucial
- The available packages are versatile
- Optimization slightly improved our results
- Future work: need more datapoints at higher redshifts